Principles of Engineering System Design

Dr. T. Asokan
Department of Engineering Design
Indian Institute of Technology, Madras

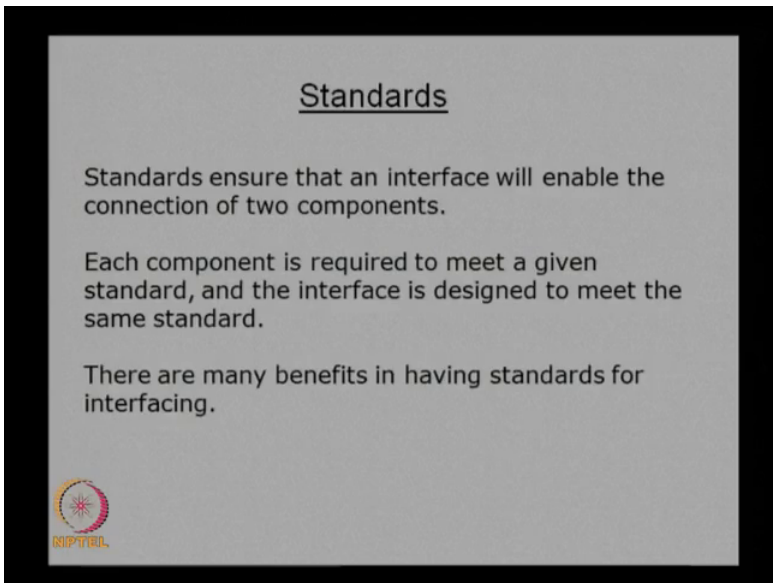
Lecture – 17
Interface Standards and Design Process

Dear friends, welcome back to one more session on engineering system design. In the last lecture we discussed about the interface design how do we actually identify the interfaces and how do we design the interfaces for the system to communicate between with the external system as well as the communication within the subsystems. And we discussed about various a types of interfaces I am basically looking at the architectures available like a message passing, shared memory, then local area network.

And then we saw few a architectures in the local area network itself like a start network or a mesh network. And in this lecture we will try to see the existing standards for interfacing. As you know that we need to have some kind of standardization and interfaces interface design because this will help us to have the system with more flexibility and more reliability as well as we can reduce the cost of developing the system also.

So, here we will look at those standards existing for a communication system especially the communication architectures for engineering systems.

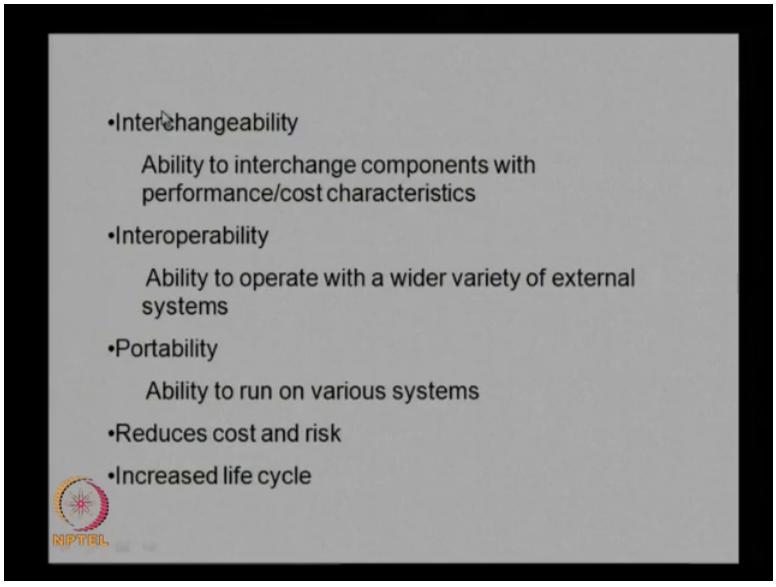
(Refer Slide Time: 01:30)



As I discussed in the last class the standards basically will ensure that an interface will enable the connection of two components. So, the basically we need to ensure that there is proper communication between two components. So, we can use the standards to ensure this communication. And whenever we have a communication between two components we need to ensure that the each component is require to meet given standard and the interface is designed to meet the same standard. So, we need to have the component meeting some standards and the interface also meeting some specified standard in order to accept these components to be part of the communication network.

And as I mentioned in the last class there are many benefits for having standards for interfacing when you say particular standard or we standardize the a interfaces we get many benefits some of them we saw in the last class basically the interchangeability of components.

(Refer Slide Time: 02:22)



So, when we have these standard components and standard interface it is possible for us to interchange components with performance or cost characteristics.

We can look at the components which are having a different performances or different cost and then try to replace these parts in order to get better design or in order to have a low cost design or a high quality design. So, unless you have this standard we will not be able to choose the components with a various options like the performance or cost. So, that is the first benefit of having a standard. For example, when we have a common

standard for different components then a manufacturer is able to replace one hard disk or a ram or a processor for another one for increase the performance or reducing the cost.

So, this interchangeability is an important feature for a many of the systems. So, other one is interoperability where the ability to operate with a wider variety of external systems. So, here also if a system has to interact with the external systems and we need to have wider options otherwise you will be limited with one or two options and then that may not be a cost effective way or a reliable way to do the system design. So, if you have the a standard it is possible to have a interoperability of a components with the system. For example, if you have a parallel port and a serial port in a computer then it is possible for us to connect any serial printer or a parallel port printer with the computer or you have a USB interface then it is possible for us to have a various components attach to the computer, so that kind of flexibilities offered by having this kind of standards for systems.

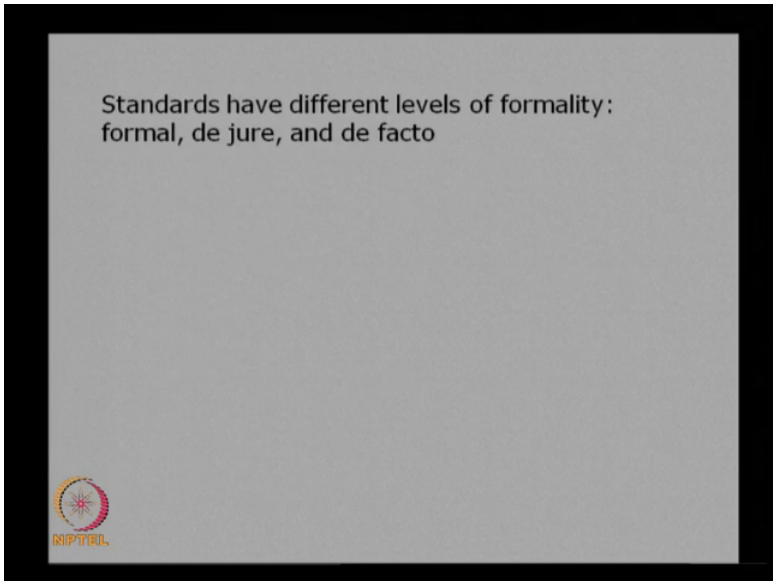
The third one is the portability that is the ability to run a on various systems. So, if you have a system and it is to run on different environments or different situations then if you have the standard followed then it is possible to port the system to another one. Examples are basically the software systems were actually can actually port into any particular operating system if you develop software and it can be operated on different platforms then it provides you a portability of the system to have actually operate under various situations, various under various situations in on various systems also.

And as you know it actually reduces the cost of the system basically because the cost can be reduced if you can chose a component which actually better than the existing one or it has actually a cost benefit then it is possible to chose that component because the interfaces are same. So, if you have the same interface and the same standard interface it is possible to reduce change the component to reduce the cost and then actually you can reduce the risk also by providing alternative options in the system even if one fails which is possible for us to replace with another one.

So, that way we can reduce the risk also and all these actually leads to an increased life cycle of a component. So, even we have portability and reduce cost and risk all these lead to an increased life cycle of the system. So, as you can see a following a standard or

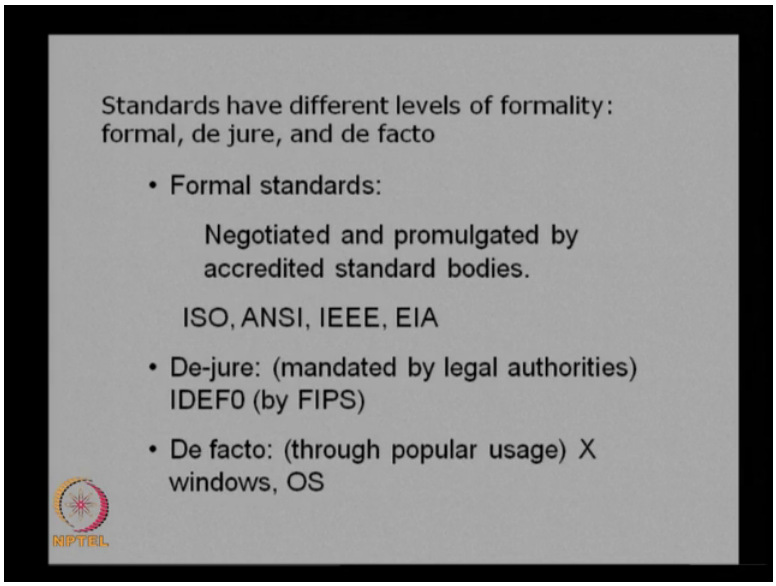
having some standards for interfaces actually helps the system designer to have a better system with the low cost as well as a low risk and with a better life cycle.

(Refer Slide Time: 05:54)



So, let us see what are the standards or how the standards are classified into under various categories. As you can see here there are formal standards, then de jure standards and de facto standards. So, these are actually coming from different origins. So, formal standards are the one which actually formally accepted by agencies or standard accredited agent bodies the negotiated and promulgated by accredited standard bodies like ISO, ANSI, IEEE, EIA etcetera.

(Refer Slide Time: 06:13)



So, these are the formal standards formally accepted by the community that is a communication standard or a hardware interface standard there are standards approved by the agencies like ISO and ANSI and IEEE. If you follow this standard then we can always ensure that all the manufacturers or all the system providers if follow these standards or what we can ensure that what are may be our system design the interface need to follow the standards. So, it always ensure the interchangeability and interoperability and other features. So, these are the formal standards already accepted by the agencies.

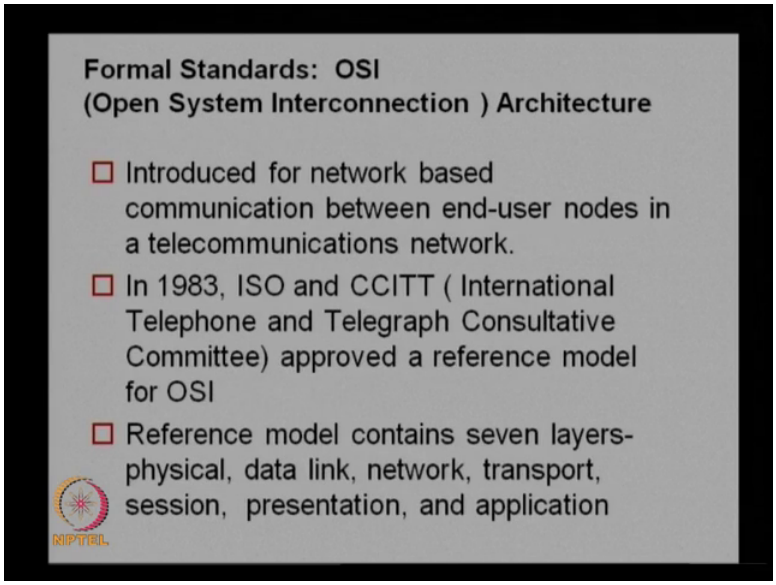
The other one is a de jure standards these are mandated by legal authorities. So, these are legally bonds everyone has to follow this standards the formal standards are not legally bonds, but if you follow that one there are lot of benefits. But in the case of a de jure standards these are a legally bond and we need to a everyone has to follow these rule especially when it comes to safety and other aspects these de jure or legal rules or the legal standards need to be followed.

So, like IDEF0 is a legal standard for a system development. So, like that there are many standards legally bond and every designer need to follow this legal standards and the last category is the de facto standards. De facto standard is basically becoming a standard because of the popular usage if a large community large a group of people start using this a particular standard then it becomes a de facto standard. So, they are not legally bond or

they are not formal standard it becomes a standard adopted by the community. And most of the cases every standard actually start as a de facto standard and then it is getting formalized as formal standards or de jure standards.

Like operating systems and x windows and this kind of standards are de facto standards which are not bound by the any legal a restrictions or formal standards. But they are become standard by use of particular standards by a group of people or a community of designers. So, these are the three classifications of standards in terms of its origin and in terms of its application and the legal validity. Let us go through some of this standards and then see how they are actually applied or how they are implemented in the real situations.

(Refer Slide Time: 08:59)



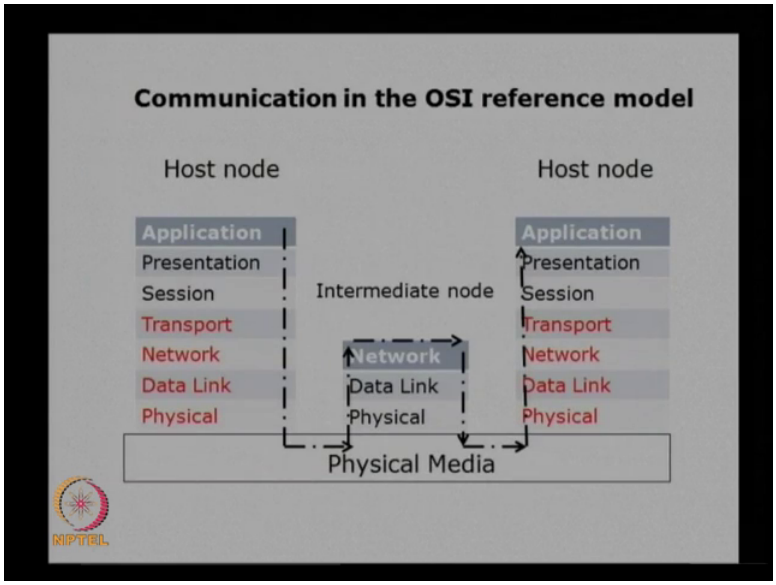
See one of the formal standards is known as open system interconnection this is a formal standard basically a used for a network based communication between end user nodes in a telecommunication network.

So, here this is basically for communication networks. So, when you have a many nodes in a communication network. So, this kind of a standard is formally adopted known as the open system interconnection standards. And this was the work on this standard was started in some sometime in 1977 and there were few de facto standards existing at that person and at that time some of them are proprietary and after lot of negotiations and a

discussions. So, these standards were combined together to make into a formal standard of a standard known as OSI.

So, it was in 1983 ISO and CCITT that is international telephone and telegraph consultative committee approved a reference model for OSI. So, that the top layer is a not reference model for open system interconnection. And this model contains 7 layers physical layer, data link, network, transport, session, presentation and application. So, these were the 7 layers identified for the reference model for OSI. And each layer has what its own responsibility and its own functions and communication between any two nodes need to follow these layers and then there are standards and functions identified for each layer for making communication between two nodes.

(Refer Slide Time: 10:37)



These are actually shows the communication in an OSI reference model. As we can see here there are host nodes. So, these are the host nodes the between these nodes if want to have a communication in a system. So, you will be having multiple nodes like this want to have communication between this node and this node and then these are the 7 layers identified for the communication node. So, every host nodes should have all these 7 layers this three are known as the upper layers and these are known as the lower layers. So, the first 4 are the lower layers and the next 3 are the upper layers.

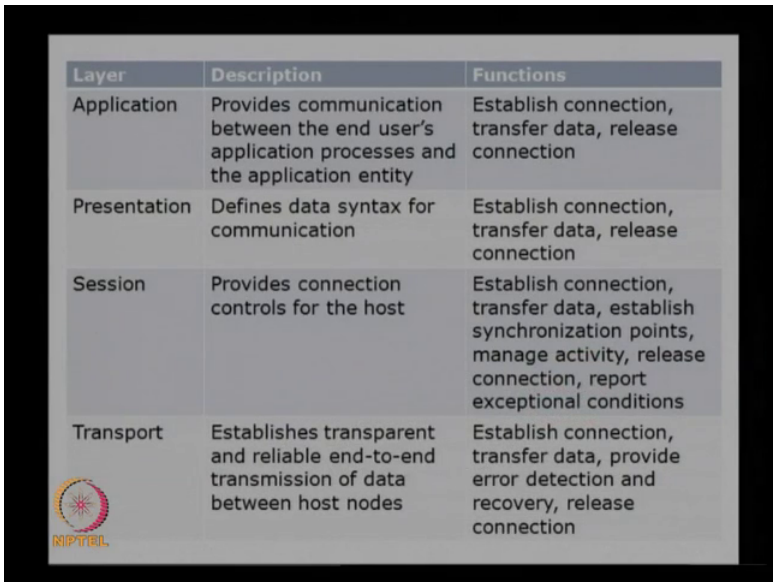
And between these host nodes we will be having intermediate nodes can have a one intermediate node or more than one intermediate node can be there. And every

intermediate node should have these three layers the first three layers that is the physical data link then network layers are there for the intermediate nodes. So, here which is shown that there is only one intermediate node between two host nodes, but that may not be the case always you can have a multiple intermediate nodes through which the communication takes place and the data is transferred from this one host node to the other host node. As we can see here this application presentation session transport network data link and physical are the layers and the physical media is the media through which the communication really takes place. And these responsibility or the functions of these nodes basically these layers basically to ensure the communication from this application to this application.

So, this application is basically the one which actually stars communication and this application needs to communicate with this application in the other nodes. So, the data comes from here and passes through the physical media and through the intermediate node and reaches the application nodes.

Other than the physical layer the other layers have the responsibility to make the communication between these layers. So, the application has to communicate with a presentation and that has to establish the communication it has to transfer the data and it has to disconnect the communication similarly presentation has to communicate with session, session has to communicate with transport, transport has to communicate with network, and network has to communicate with data link, and data link will communicate with physical and the physical layer will actually transfer the data to the physical media and then it will go through the intermediate node and come back to the application. So, this is the way in which the communication takes place in the OSI reference model.

(Refer Slide Time: 13:07)



If you look at the functions of this each layers we can identify that the application is the highest level the seventh layer in this one and the description of this particular layer is basically to provide communication between the end users application processes and the application entity. So, the application entity is the one which actually passes the information, so that it provides communication between the end users application processes and the application entity.

So, if the communication need to establish between the application entity and the end user so that the main job of the application is basically to provide this communication. And the functions of this layer is basically to establish a connection, transfer data and release connection. As I mentioned earlier every layer has these three functions as a common except for the last layer that is the physical every layer has to do the functions of establish connection ram, transfer data and release connection.

And the next layer presentation basically defines the data syntax for communication. So, the application has to transfer the data establish the connection and transfer data to the presentation layer and it is a presentation layer which actually defines the syntax for data. So, what kind of syntax to be used for communication is decided by the presentation layer. Again the functions here are establish connection transfer data and release connection. So, as I mentioned a seven function with these also has to do, but in addition it will actually design about define the data syntax for communication also.

The next layer is session where it provides connection controls for the host. And basically the control of the connection between the layers that the host will be provided by the session and establish connection transfer data establish synchronization points, manage activity, release connection, report exceptional conditions are the main functions of the session layer.

So, apart from this establishing connection a transferring data and release connection; these layers has an additional job of establishing the synchronization points with the other layers, and then manage the communication activity and a report exceptional conditions. So, if there are any exceptional conditions are arising that has to be reported to the application immediately when whenever that arises. So, the function of this layer is basically apart from establishing connection and transferring data we will look at the synchronization as well as the exceptional conditions.

The next layer is transports here it establishes transparent and reliable end to end transmission of data between host nodes. So, the transport layer is basically responsible for transporting the data it establish a transport and reliable end to end transmission of data between host nodes. So, this actually takes care of the transmission of data from one node to another node. Of course, it has to communicate with the session for the data if the data is coming from this layer. So, it will establish a connection with the layers and then transfer data it will provide error detection and recovery and release connection. So, apart from communication it has got the additional function of error detection and recovery and that is the function of the transport layer. So, it establish the connection and then transfer the data to the other nodes. So, the basic transmission is the responsibility of transport layer.

The other layers are the network layer here it determines the establishment of connection and handles the routing. So, the establishment of connection is the responsibility of a network that irrespective of the type of network it has to establish a connection. So, whatever maybe the network architecture it will establish a connection and handles the routing of a data, so to which node it has to how it has to be transmitted what kind of data to be used will be decided by the network layer. And here you can see establish connection, transfer data, perform multiplexing, provide error control and release connection. So, establishing connection and release connection as the basic one, but apart from that it has to perform the multiplexing and provide error control. So, these are the

additional functions it has to perform other than establishing connection and transferring data.

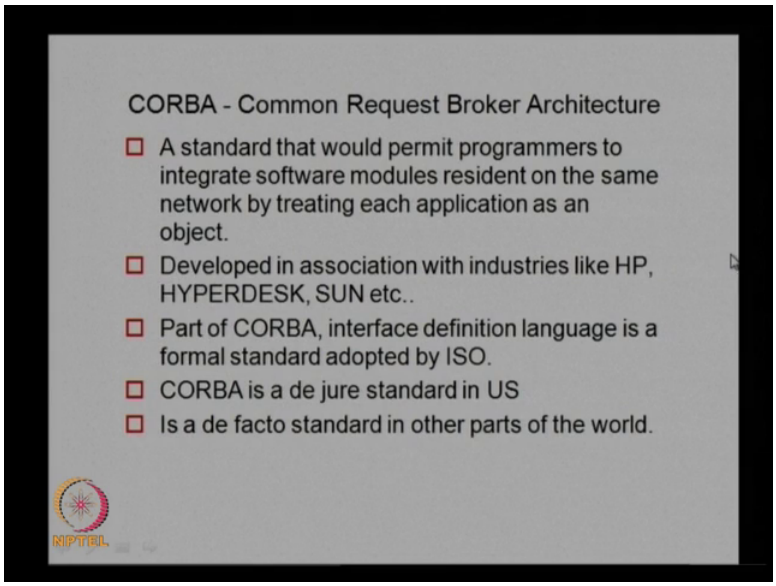
The next layer is the data link. Here it establishes a reliable transmission on the physical layer. So, data link is responsible for transferring the data to the physical layer that is a reliable transmission on the physical layer its responsibility of the data link. Here this establishes connection, negotiate quality of service, it transfer data, provide flow control, reset connection and release connection. So, these are the functions. So, it has to negotiate the quality of service is the quality of the service is basically the responsibility of the data link. So, it will negotiate in terms of rate and the speed and other information about basically if you look at the integrity of the data and then try to ensure that the integrity of the data is not compromised because of various reason and that is the responsibility of the data link and it provide a flow control also and then reset connection and release connection. So, these are the functions of data link.

The last one is a physical layer. So, it defines how the physical network is accessed in order to provide with transparent transmission. So, now, the next one is from here the physical layer it means the data will be going to the physical link, from there, it has to define how this physical network is accessed in order to provide bit transparent transmission. So, that is the job of physical. The main functions are determine a presence of a signaling pulses determine timing of signaling pulses. So, these functions need to be provided by the physical layer in order to ensure that the physical network is accessed and then a bit transparent transmission is taking place. So, these are the main layers in the OSI and the functions of different layers.

As I told you the every node every transmission node the host node as well as the receiving node, need to have these a physical layers and then it will be interacting through the intermediatory node where there will be only a three layers. So, there can be multiple intermediary nodes, but whatever the number of intermediatory nodes the host node as well as the receiving nodes should have all these physical layers and their functions are already explained.

So, based on the responsibility of the functions of each layer the communication will take place. So, that is the standard a formal standard for communication between two nodes in the OSI architecture.

(Refer Slide Time: 19:39)



Let us look at another standard which is known as the common request broker architecture this is basically a standard used for software development. So, when you have different software modules used for a particular system as for interfacing the systems or a communication between the systems or subsystems, this CORBA is followed as a standard. So, this is a standard that would permit programmers to integrate software modules resident on the same network by treating each application as an object.

So, this actually helps the programmers to integrate different software modules treating each application as an object. So, each application will be considered as an object and then how do we actually integrate this object or these modules is defined in the architecture the common request broker architecture of standard.

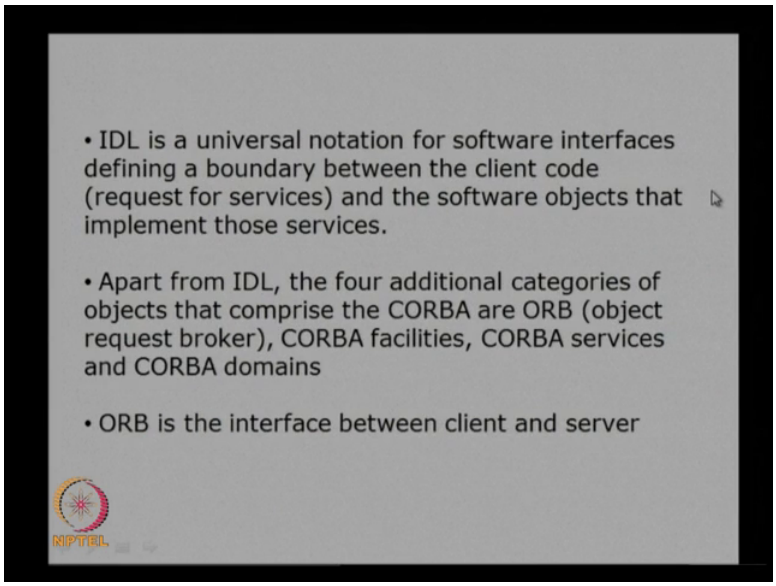
This was developed in association with industries like HP, Hyperdesk, SUN and so on. So, there are many industries participating in the development of this kind of a standard this CORBA was developed in association with a HP, Hyperdesk and SUN. And a interesting aspect of a CORBA is that it actually is part of the CORBA that is the interface definition language or a IDL, it is a formal standard adopted by ISO.

So, part of a CORBA is a formal standards and it is a de jure standard in the US. So, in the US everyone has to follow and this standard because it is as per the legal its mandatory as per the legal standard a CORBA is to be followed in software development and especially for when you develop different modules and want to integrate these

modules. But it is a de facto standard in other parts of the world. So, though it is not the mandatory in other parts of the world and it is not a formal standard many of the developers follow this standard in other parts of the world. So, it becomes a de facto standard.

So, we can say that it is a combination of all the three type of a standards that is the formal standard and de jure standard and de facto standard. So, part of CORBA the IDL is an ISO standard and CORBA is a de jure standard in the US and in other parts of the world it is a de facto standard, most of the software developers tried to follow this CORBA standard when developing the software's modules.

(Refer Slide Time: 22:01)

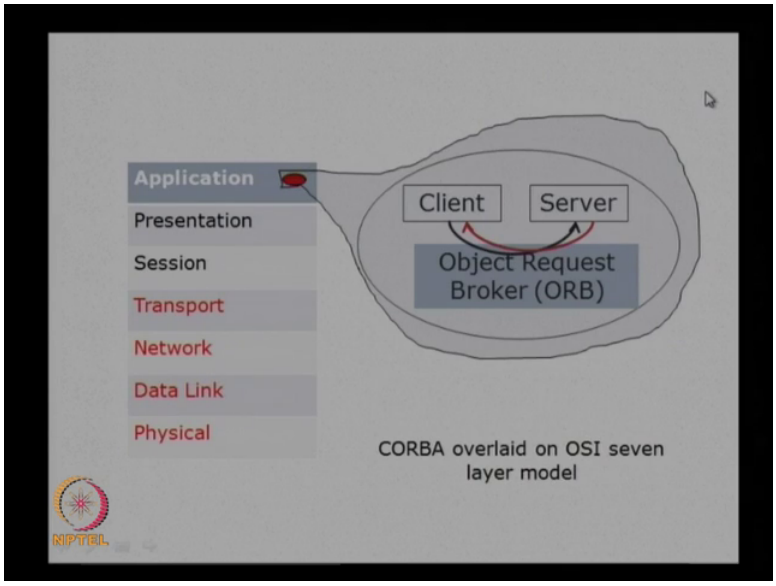


We will just look at what are the important features of this IDL. As you can see that IDL interface definition language is a universal notation for software interfaces defining a boundary between the client codes and the software objects that implement both services.

So, IDL as you can see it is a formal standard and it is a universal notation for software interfaces defining a boundary between the client code that is the request for services and the software objects that implement those services. So, it actually provides an universal notation for the interfaces. Client code the request for services as well as those for implementing these services they can actually follow this codes the universal notation when they develop the software. So, that actually is the interface definition language.

But apart from IDL and there are 4 additional categories of objects that comprise ORB which is the object request broker, and then CORBA facilities, CORBA services and CORBA domains. So, these are the 4 categories additional categories of objects that is needed in a in the CORBA language, CORBA standards. So, IDL forms the a basis for the interface and the other categories of objects like ORB and facilities services and domains become additional objects. ORB the object request broker is the interface between the client and server. So, whenever there is a client and server and you want to have an interface then ORB is the interface object request broker is the interface in this case.

(Refer Slide Time: 23:34)



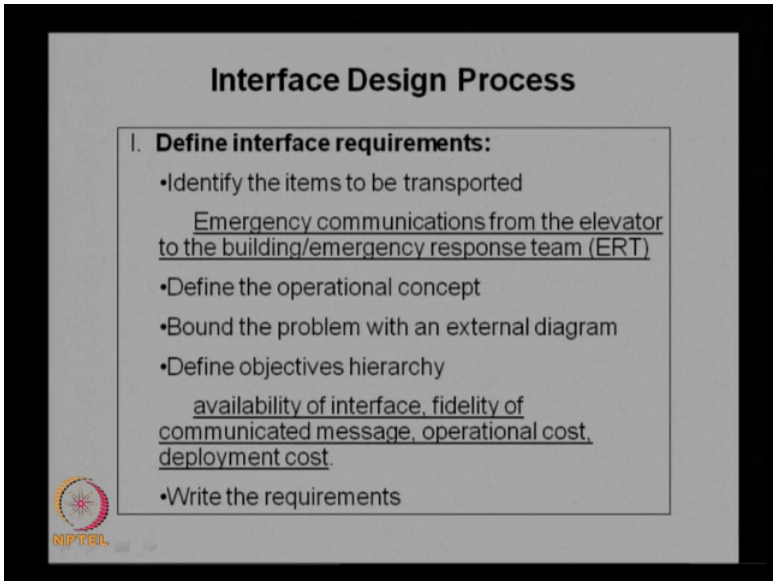
Now, how this is implemented in the system is shown over here. As you can see the every application will be having these interfaces. So, application is the one which actually tries to connect to the next node. So, this application will be having their client and server and then object request broker is either intermediate or the one which actually connects between the client and server as shown here. So, there is communication between client and server and this is actually controlled or the standard for this communication is the object request broker.

So, this actually shows the CORBA overlaid on an OSI 7 layer model. So, these are the 7 layer model. So, this layer application layer will be having the CORBA standard or the object request broker in the application module. Basically the standard for

communication between the client and the server, so that is the object request broker which actually interfaces client and server. So, this is how the ORB is implemented in the OSI standard. So, that was about the standards used in the system design. So, we saw a few formal standards based on ISO and other predicted bodies and then we saw some of the one standard like this is a CORBA which is used for the software.

The first one that OSI was for a communication networks and CORBA is used mainly for the software modules. So, one is formal standard and CORBA is actually partly formal standard and then it is a de jure standard in the US and the de facto standard in the other parts of the world. Those are some of the standards being used for developing the interfaces. We discussed about the interfaces different kinds of interfaces and then a different standard for interfaces. So, let us look at what is the process of designing an interface or a how interface is designed by the design engineers or what are the stages through which they go in order to develop the interface.

(Refer Slide Time: 25:18)



So, the first stage is basically to define the interface requirements as you know when we do a system design we try to develop the system requirements in the same way interface design process also can be considered as a developing as a process where we try to identify the requirements. Basically we identify the items to be transported. So, what are the items to be transported what kind of data needs to be transported between the systems or the subsystems that need to be identified this is an example.

So, one example may be the emergency communications from the elevator to the building or emergency response team. So, this could be an item to be transported. So, when we look at the design we identify the items to be transported. For example, the elevator case study you can see the emergency communication could be one item to be transported from one system to other system or one subsystem to another subsystem. So, like this we can identify for every interface we can identify what are the items to be transported. So, that is the first stage in the interface development, identify all the items to be transported.

The next one is define the operational concept. So, what kind of a system or what kind of a an interface can be designed for this particular data that is the operational concept. We can have various concepts for operation then we will identify which one is the best one and accordingly we define a particular operational concept for that particular interface. And then bound the problem with an external diagram, so we look at that particular part alone the interface alone and then bound the problem with an external diagram. So, what are the other systems will be which will be interacting with this interface and what kind of a interaction are needed between these interfaces that will be identified in the external diagram.

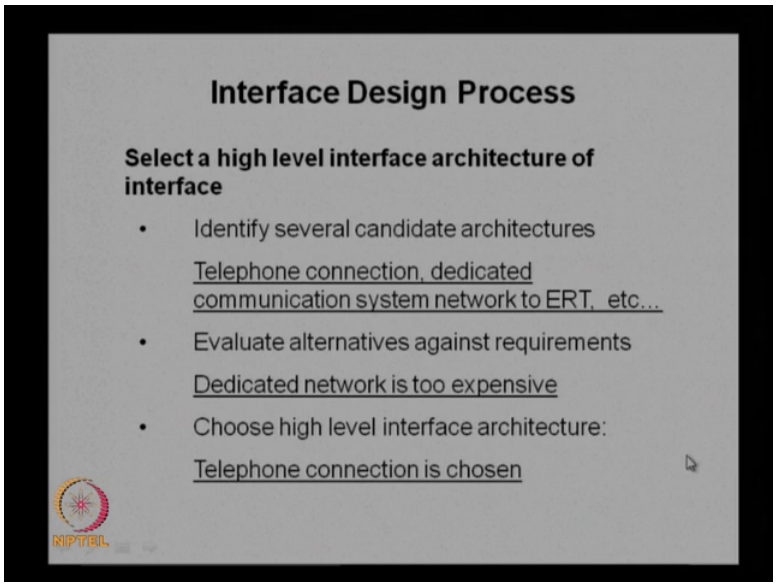
So, once you identify the data to be transferred and then you have operational concept then we go for the external system diagram for the particular interface. And then define the objectives hierarchy, so here again you will be having many objectives this one is we need to transfer the data, then we will be having the transferring the data at a particular rate, then you will be having the cost objectives. So, we will write down all the objectives for the interface and then a hierarchy of these objectives we need to find out which one is more important whether the transmission rate is more important or the quantity of data to be transfer is more important or the cost of operation is more important or the cost of the whole system is more important. So, accordingly we develop an objectives hierarchy.

Like availability of interface fidelity of the communicated message operational cost deployment cost. So, these are some of the objectives. So, then we will give a value for these objectives as we discussed in the previous lectures about objectives hierarchy we will identify the hierarchy of these objectives based on the requirement of the particular

system. So, the system designer will be knowing what is more important in this case and accordingly he will prepare an objectives hierarchy.

And then once you have this identified then we will write down the actual requirements of the design process. So, we start with the items to be transported, then we have an operational concept, then we will have the external diagram, then the objectives hierarchy once you complete all this, then we will be having all the requirements identified and then write the requirement as we discussed in the originating requirements document similar way we can write down the requirement for the interface also. The interface system requirement will be separately identified.

(Refer Slide Time: 28:51)

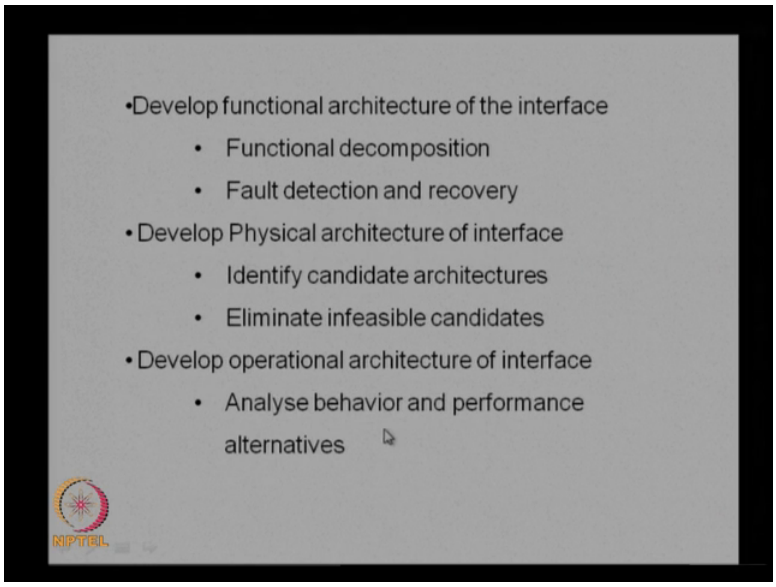


And then select high level interface architecture for the interface. So, you need to have the architecture for the interface. So, we will select very high level interface architecture for the interface or what kind of architecture can be implemented based on the requirements identified. So, here we can actually identify several candidate architectures. Basically if you want to have an emergency communication we can think of a telephone connection, you can think of a dedicated communication system network, to emergency response team or you can have kind of a buzzer which is giving output to some other sources some other systems. So, we can have many candidate architectures for the interface. So, that is the selecting high level interface architecture for the interface to be developed.

And then we need to evaluate these alternatives against the requirements. So, we have a requirement document which is which actually obtained from looking at the requirements of the interface in terms of the data a item to be transported and other features, based on that you have to evaluate the alternatives against the requirements. So, we have many alternatives identified and evaluates these alternatives against the requirements.

And then based on this you will be getting some idea like the dedicated network is too expensive in the case of elevator system. So, you can actually eliminate that one and probably you can choose the telephone connection as a possible option. Again these are examples the actually depends on various other factors. So, the designer need to look at all these alternatives and then decide what is the best one for this particular system. And then choose high level interface architecture based on this once you evaluate the alternatives we will get a high level interface architecture. So, in this case since this is an expensive one. So, you can go for a telephone connection which will be the right interface for communication from the elevator to the emergency response team. So, that is the way how you choose high level interface architecture.

(Refer Slide Time: 30:46)



And then we go for the functional architecture of the interface as we discussed earlier the functional architecture will tell you what kind of functions to be provided and what are the features to be need to be incorporated in the system in order to meet the

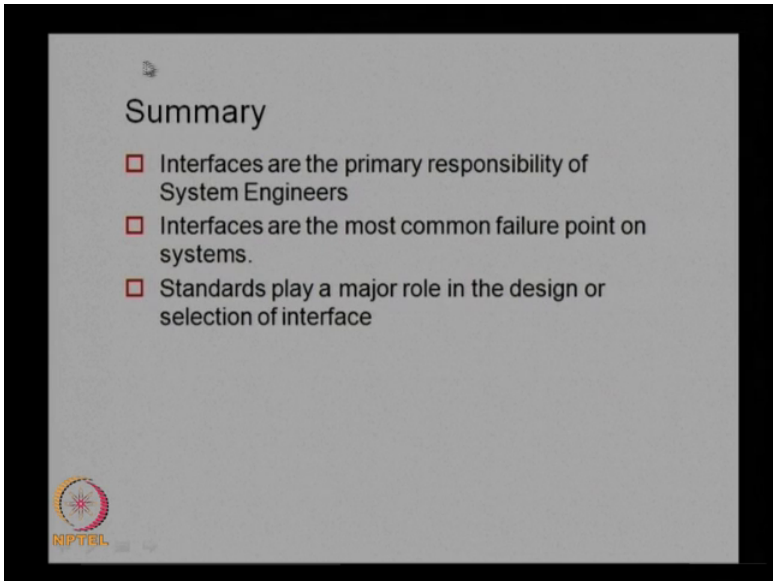
requirements. So, there may be many functions needed to meet the requirements, so will through a functional architecture all those functions can be identified.

So, basically we use the functional decomposition methods and we add the fault detection and recovery mechanism or the functions also along with the actual functions or the required functions. So, all this will comply I mean will add to the functional architecture, so you will get the complete functional architecture based on the decomposition as well as the fault detection and recovery. And once you have the functional architecture you go for the physical architecture of interfaces. A physical architecture is basically identifying the candidate architectures for the identified functional decomposition and then eliminate infeasible candidates.

So, you can get a multiple architectures based on the functional architecture and then eliminate the infeasible candidates and choose a physical architecture which is a most feasible for the particular interface. And the next stage is basically making an operational architecture of interface that they analyze the behavior and performance of alternatives. So, you look at the allocation of functions to the candidate architecture or the candidate components and then look at their interfaces and the requirements and based on this we will have an operational architecture of interface.

So, it basically we will be looking at the allocation of functions and the behavior and performance of the alternatives that will give you an operational architecture.

(Refer Slide Time: 32:29)

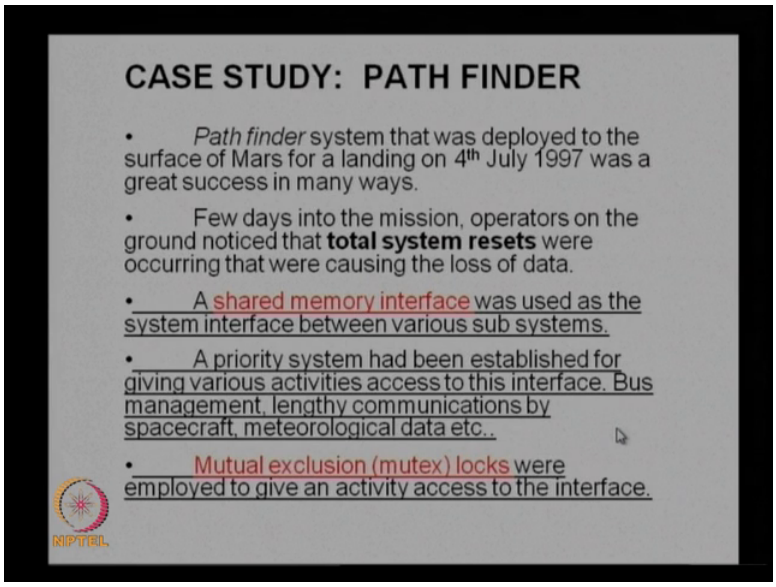


So, to summarize this procedure whatever we discussed about the system design also the interface system design you can see that the interfaces are the primary responsibility of system engineers. So, the system engineers need to ensure that they provide the necessary interface for the system and they are very sensitive area and they may lead to the failure of the system. So, it is necessary for the system engineers to look at the interfaces and design them properly.

And then the interfaces of the most common failure point on the system and the standards play a major role in the design or selection of interfaces. The duty of the system engineers to ensure that they provide the proper system interface and they develop the interface architecture based on the a standards existing and the standard methods available for the interfaces and need to ensure that these systems actually the interfaces provide a reliable communication, a reliable interface between the system and external systems as well as the subsystems. So, that is about the design of interfaces for engineering systems.

When we discussed when we started this design of the engineering the interface of the systems we discussed about a case study. Now, you can actually have a revisit to this case study and then see what are the points which were very important when we discussed about that case study and how the system designed the interface design helps to eliminate those features.

(Refer Slide Time: 34:00)

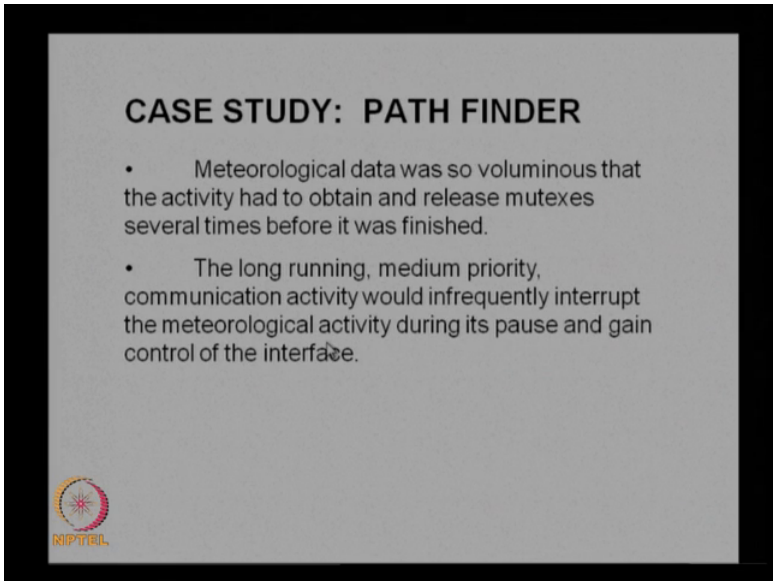


So, this was what we discussed in the beginning about the path finder which was deployed 1997 and there was some total system resets and it was being employ or it was use observed that there were total system resets happening.

And on the future was that a shared memory interface was used as the system interface between various subsystems. So, we discussed about the shared memory interface and its features similar shared memory interface was used in the pathfinder also to have the communication between various subsystems.

And then a priority system had been established for giving various activities access to this interface bus management lengthy communications by spacecraft meteorological data etcetera and here the mutual exclusion locks were employed to give an activity access to the interface. So, you have a shared memory interface and a mutex which actually allow a particular activity to get access to the shared memory interface.

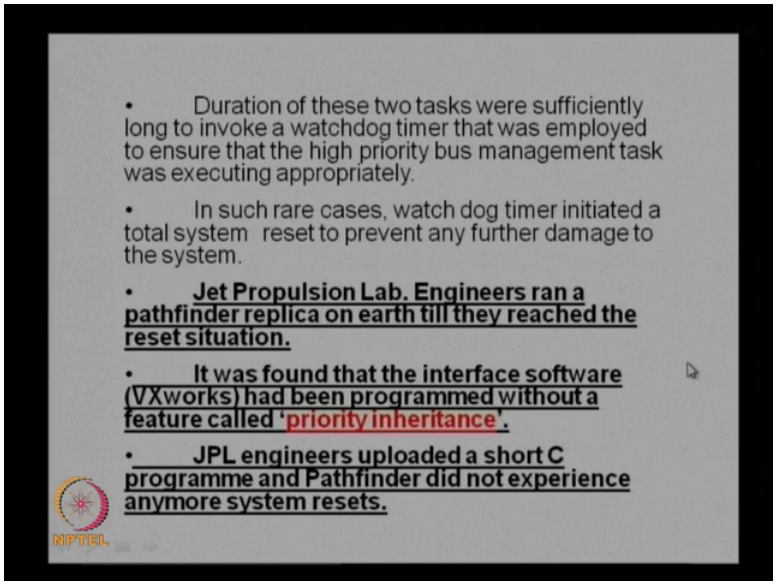
(Refer Slide Time: 35:04)



And there were two activities which actually were taking a long time to complete basically the meteorological data which was so voluminous that activity had to obtain and release mutexes several times before it was finished. And the long running medium priority communication activity would infrequently interrupt the meteorological activity during its pause and gain control of the interface. So, these two activities were take long running activities and they were actually competing each other to take control of the

network, mutexes and that was actually creating a problem in terms of the watchdog timer.

(Refer Slide Time: 35:33)



So, there was a watchdog timer that was employed to ensure that the high priority bus management task was executing appropriately. So, the watchdog primer was there to ensure the high priority bus management was executing appropriately and because of the other two activities which were actually competing to get the mutex the control of the network were actually causing an error in the watchdog timer and the watchdog timer was finding that something is wrong in the system and it was trying to reset the whole system to avoid any possible damages.

Then of course, we discussed that jet propulsion lab engineers ran a replica on earth and then identify the problem and then they incorporated a priority inheritance in the interface software which was VX works. So, it was programmed without a feature called priority inheritance and then they changed this code and added a few lines of seek program to the system and got the priority inheritance, incorporated into the system and then run the path finder and then found that it is working perfectly well.

So, this actually shows that you have a various interface procedures or we can actually have a interface design in various ways, but important point is that we need to ensure that the communication between the a subsystems or the external systems are not affected because of any other unexpected feature. So, it is the responsibility of the design the

system engineers to ensure that they a take care of the data integrity as well as the other processes parallelly going on in the system and ensure that there are sufficient features incorporated in the system to ensure the system works perfectly well and there is no failure of the interfaces.

So, that actually concludes our interface design process. The next one is basically looking at the qualification of the engineering system or basically looking at the verification and validation of engineering system, how do we ensure that, what we designed actually meets the requirement that is the qualification of a validation and verification of engineering system or we call it as a system qualification procedures. We will discuss this in the next lecture till then good bye.